# 

# Tarea 1. Programación de una solución paralela

## Liga del repositorio en GitHub

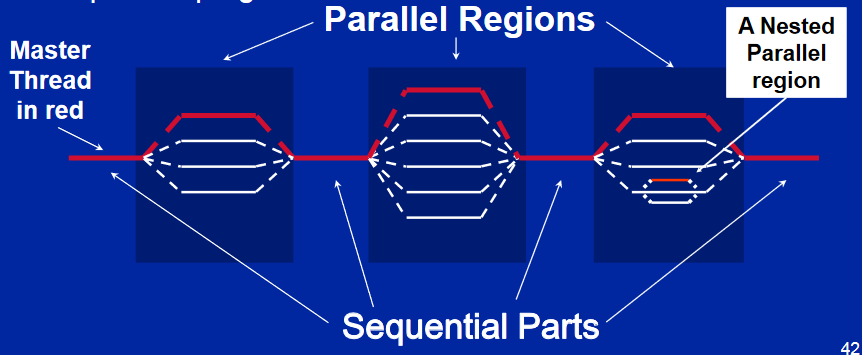
En el siguiente repositorio se puede consultar la presente documentación y el código del proyecto:

[Liga](https://github.com/flynnGodslayer/SolucionSumaArreglosParalela_Victor_Avila).

(https://github.com/flynnGodslayer/SolucionSumaArreglosParalela\_Victor\_Avila)

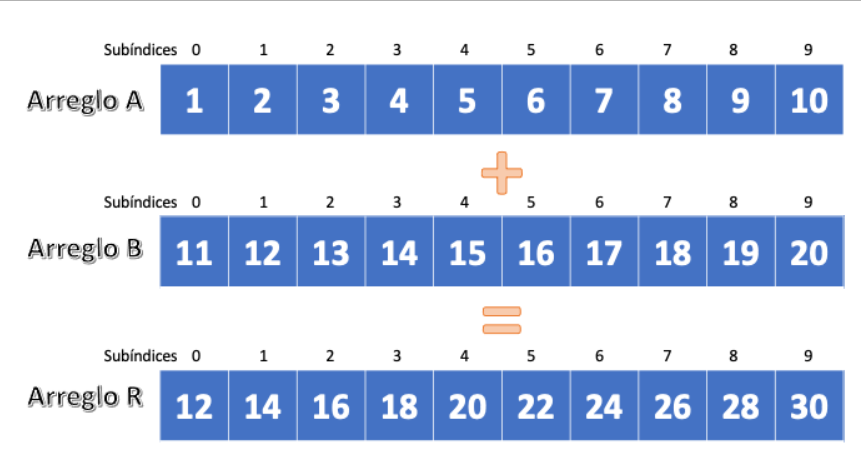
## Introduccion

Open MP es un modelo de programación en poaralelo que permite llevar a cabo una configuración de ejecución de los programas escritos en C++ por medio de hilos (threads). Hace uso del modelo fork-join en paralelo, donde un hilo maestro (Master thread) ejecuta un equipo de tantos sub-hilos como necesite.



En esta practica, se desea resolver el siguiente problema:

Piense en una suma de dos arreglos como la mostrada en la siguiente imagen, donde tenemos dos arreglos llamados A y B de 10 elementos cada uno y deseamos crear un tercer arreglo con la sumatoria de cada uno de los elementos en los mismos subíndices.



Esto es bastante sencillo en la programación, pero el tiempo que tomará en calcularse el resultado depende directamente de la cantidad de elementos que tienen los arreglos y la velocidad del procesador en la que se ejecutan las operaciones. Las sumas individuales de los elementos se realizan una por una en orden según los subíndices, entonces ¿Qué pasaría si los arreglos tienen millones de elementos?

## Solucion del algoritmo

La respuesta es que la suma de los arreglos tardaría bastante en resolverse. Entonces aquí podríamos aplicar la programación paralela para que la suma de algunos elementos de los arreglos se realicen en un thread, otra parte las controle otro thread y así sucesivamente, esto aprovechando que las sumas son independientes entre ellas.

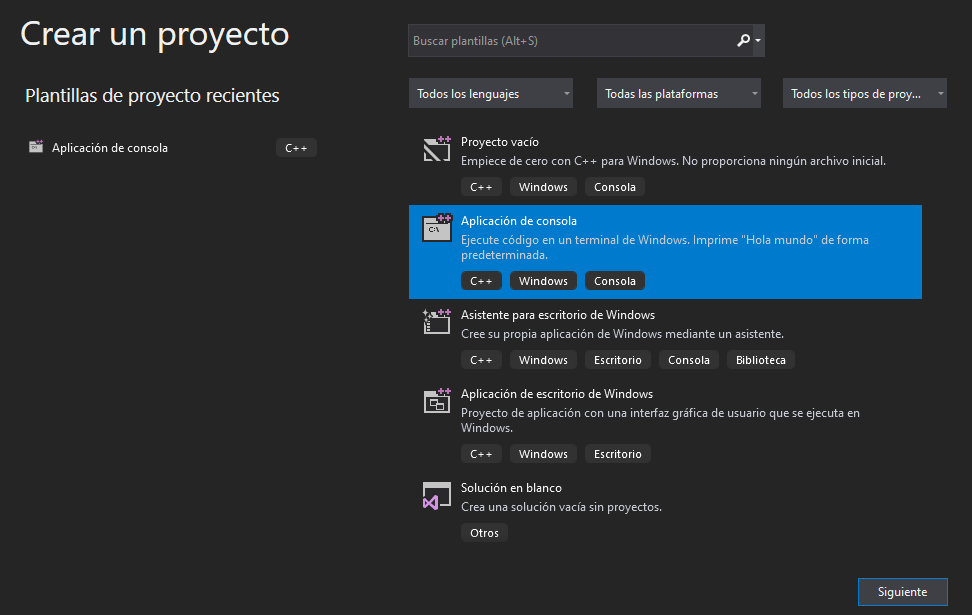
Entonces podríamos pensar en algo como lo mostrado en la siguiente imagen.



## Proceso

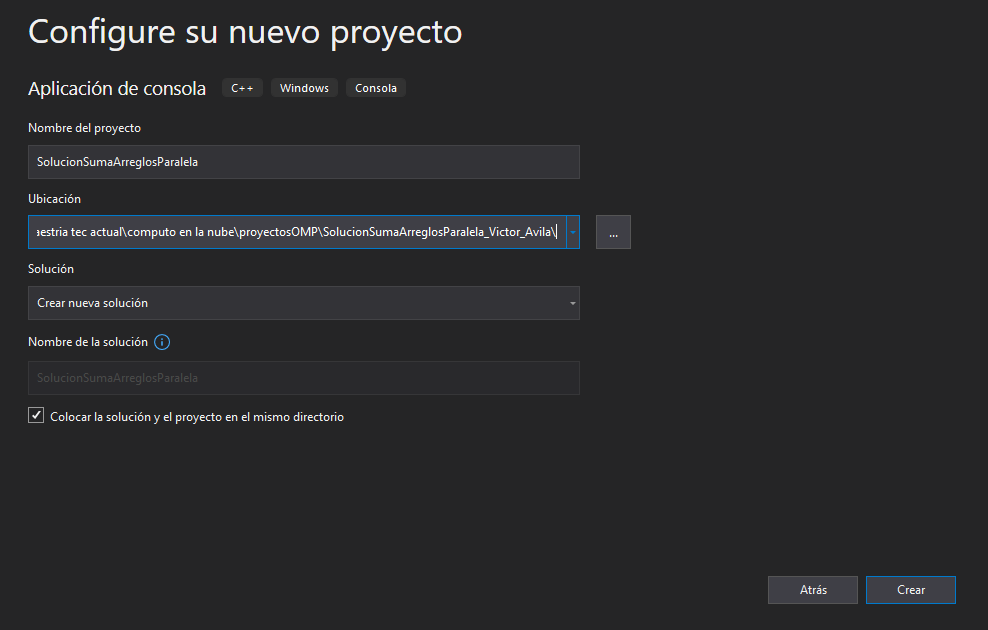
### Crear un proyecto en Visual Studio

Creamos un proyecto nuevo de aplicación en consola dentro de VisualStudio. En mi equipo cuento con la versión 2019.

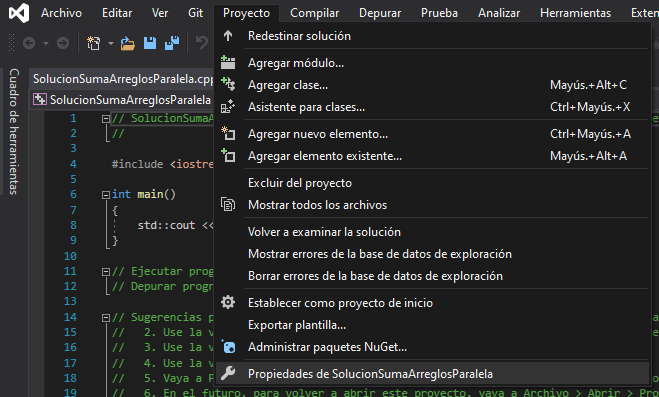


### Configurarlo para que pueda utilizar la librería OpenMP

Configuramos el nombre de la carpeta donde alojaremos los archivos a ejecutar.

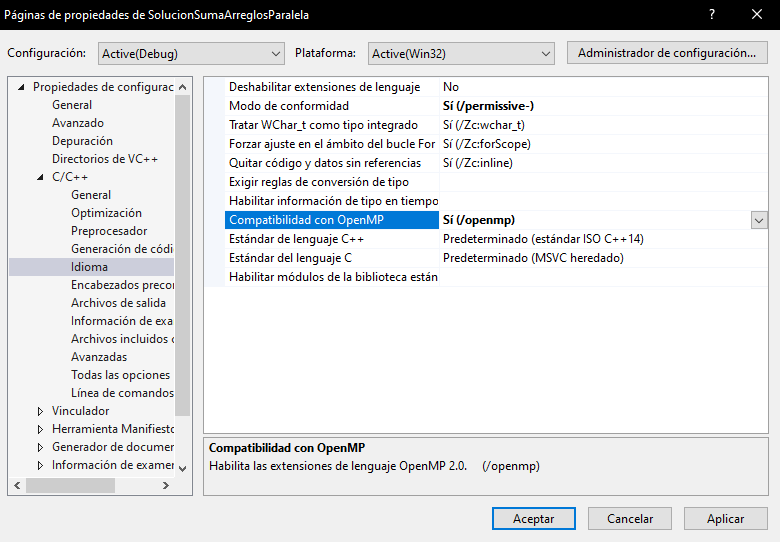


Una vez iniciado el IDE con el proyecto, toca configurar la compatibilidad de este con OpenMP. Ingresamos a la configuración de la siguiente manera:

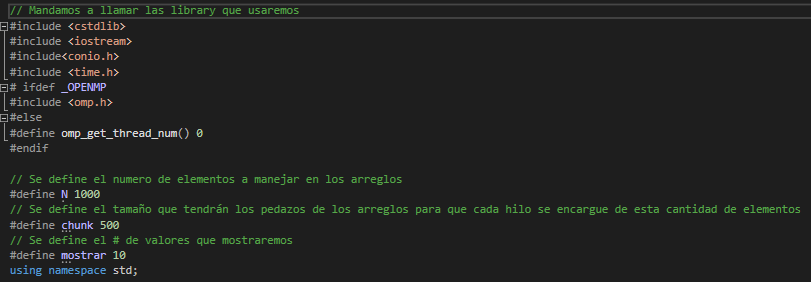


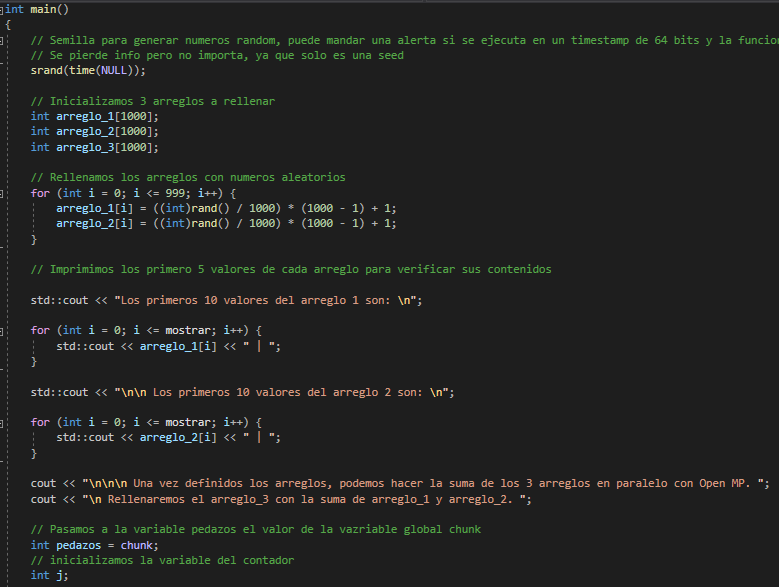


En el sub-menú idiomas, habilitamos la compatibilidad, y aplicamos los cambios:

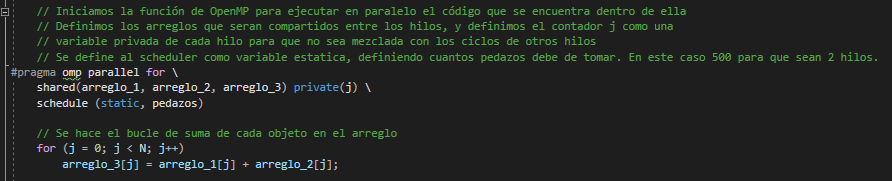


### Crear los arreglos y asignarles valores aleatorios o pedirle los valores al usuario (la cantidad de elementos puede ser de 1000 para ejemplificar el proceso y obtener rápidamente resultados)

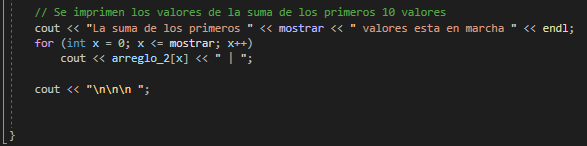




### Realizar un for paralelo a través de las funciones de la librería OpenMP

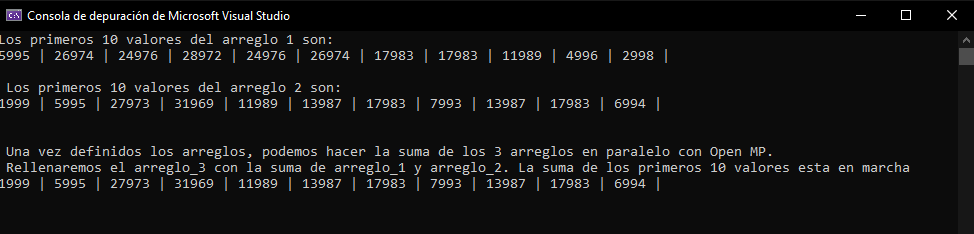


### Imprimir los arreglos o parte de ellos para comprobar que el arreglo resultante contiene la suma de los dos arreglos

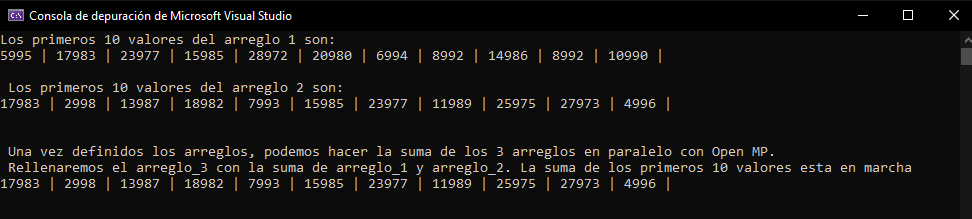


## Resultados

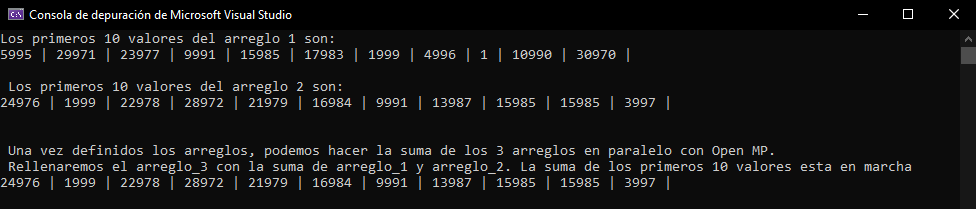
### Ejecución 1



### Ejecución 2



### Ejecución 3



Reflexion

El uso de las bibliotecas de OpenMP son sencillas cuando se usa el